AIAA JOURNAL
Vol. 35, No. 6, June 1997

Octree-Advancing Front Method for Generation of Unstructured Surface and Volume Meshes

Harlan McMorris* and Yannis Kallinderis†
University of Texas at Austin, Austin, Texas 78712

A method is described for the generation of unstructured grids for complex three-dimensional geometries including multibody domains. A combined octree-advancing front method is presented for generation of the unstructured surface mesh as well as the tetrahedral volume mesh. The unique feature of this generator is that a special octree provides the local length scales of the mesh, which can vary significantly. This octree is automatically created and eliminates the need for a user-constructed background mesh to determine the grid-spacing and stretching parameters. The grid generator is robust, is geometry independent, and requires minimal user interaction. A technique is also presented for generation of anisotropic surface meshes that result in a significant reduction in the number of triangular faces. Automatic local remeshing is also employed to improve the quality of the tetrahedral mesh. The geometry independence of the octree-advancing front method is demonstrated through both surface and volume meshes generated around different complex three-dimensional configurations. The robustness and flexibility of the method is demonstrated through generation of both an all-tetrahedral mesh and the tetrahedral part of hybrid prismatic/tetrahedral grids for viscous flows. The scalability of the generation time is shown through generation of different meshes around the same geometry with various levels of coarseness. Finally, the suitability of the unstructured grids for flow simulations is demonstrated through viscous flow simulations around a high-speed civil transport type of aircraft configuration using a hybrid prismatic/tetrahedral mesh.

I. Introduction

UNSTRUCTURED grids have emerged as a viable method for three-dimensional geometric modeling because they can cover complicated topologies easier compared to their structured counterparts.[1–5] The lack of structure allows complex geometries to be gridded in single blocks. The three main approaches of unstructured grid generation are Delaunay methods,[6–8] advancing front methods,[3,4,9,10] and octree-based techniques.[11,12] Work has also been done recently to combine the first two approaches.[13–15]

In the Delaunay approach, a satisfactory mesh requires that no tetrahedron contains a point other than its four forming vertices within its circumsphere. Advantages of the method lie in its efficiency, and the fact that a valid grid can always be achieved. However, such grids do not conform to the boundary surface triangulation without special boundary face recovery steps. Also, round-off errors in Delaunay approaches tend to be larger than in other approaches.[13]

The octree approach is based on recursive subdivision of a master hexahedron that encompasses the entire domain to be gridded. Subdivision takes place until the size of the refined octants is on the order of the local edge lengths on the boundary surface. Octree-based approaches are the fastest of the three unstructured grid-generation techniques. The algorithms are simple to code, and the generated grids provide a naturally smooth variation in cell sizes. The main problem with the method, however, is interfacing the octants with the boundary surface. Although this can be done, it often results in poor-quality meshes close to the surface. This method can also result in a very large number of elements for a given geometry.

Advancing front grid generators use the initial boundary surface triangulation as a starting point and create tetrahedra away from it. This is done by connecting each face of the boundary with either a newly formed or an already existing point to form a tetrahedron. The list of faces on the front is continually modified, and the generation is complete when no faces remain on the list. A main feature of this approach is that boundary integrity is guaranteed. This is important for hybrid meshes where the initial surface triangulation is the outer prismatic surface, and this surface must be matched exactly for a valid hybrid grid.[16–20] Maintaining the boundary integrity is an important advantage of advancing front approaches over octree and Delaunay approaches when hybrid prismatic/tetrahedral grids are required. Furthermore, there must be a smooth transition in terms of element sizes from the prismatic to the tetrahedral regions of the hybrid mesh. The advancing front method is well-suited to this task as well because it is relatively easy to control the placement of new points to ensure a smooth transition. The method can yield good-quality meshes because it allows direct control of the point placement. However, a key problem in advancing front approaches is determination of the local size and stretching of the cells in the domain. In the past, this task has been undertaken by a user-constructed background mesh.[3,4,21] Creation of the background mesh is usually very time-consuming and requires a lot of user interaction. A recent development is the use of sources to specify local point spacing,[21] but this method also requires user interaction. A new variation of the advancing front procedure that generates elements through iterative point insertion into an initial mesh requires no background mesh. This method, however, uses an initial mesh that, in specific cases, may require extensive user interaction to generate.[9]

A new method to control the size and stretching of the unstructured mesh is presented here. The main feature that is different from previous advancing front generators is that it does not require a user-constructed background mesh for determination of the grid spacing and stretching parameters. A special octree is constructed via a divide-and-conquer method of the space enclosing the entire domain. The grid spacing is then determined based on the size of local octants that form the octree. This new method provides a gradual variation in cell sizes throughout the domain. The creation of the octree is robust, automatic, and geometry independent. Another feature of the octree method is efficiency; the octree data structure provides an efficient way to search to find local size information at a particular point in the mesh. This octree is proven to be very flexible in providing local length scales that can vary by orders of magnitude for typical viscous flow simulations. Furthermore, it allows a smooth variation of the grid element sizes. Finally, the octree unifies surface and volume mesh generation.

Previous surface mesh generation has focused primarily on generating isotropic grids. By ignoring the inherent directionality in both the flow and the geometry, isotropic meshes need an inordinately large number of faces to cover a given geometry. Anisotropic meshes with triangular faces aligned with the directionality of the

Received Dec. 23, 1995; revision received Jan. 27, 1996; accepted for publication March 3, 1997.

*Graduate Research Assistant, Department of Aerospace Engineering and Engineering Mechanics. Member AIAA.

†Associate Professor, Department of Aerospace Engineering and Engineering Mechanics. Senior Member AIAA.

flow and/or the geometry offer a substantial savings in the number of faces required for a given simulation. The leading and trailing edges of a wing are good examples of areas that exhibit strong directionality and are prime candidates for anisotropic meshes. Previous methods for generating anisotropic meshes have relied heavily on specifying directions and spacings at certain points in a coarse background mesh and interpolating between the specified points[4,22] or have generated isotropic surface meshes and utilized directional refinement to produce anisotropic grids.[23] An extension to the background grid approach uses curve and line sources to generate anisotropic elements.[24] The extension of the octree method to anisotropic meshes presented here is robust and provides a simple way of specifying the desired directionality for a mesh. The method allows for smooth transition between anisotropic and isotropic regions of the triangulation as well as simple control over the direction and strength of the anisotropy.

Surface meshes as well as tetrahedral meshes for hybrid grids are generated for a high-speed civil transport (HSCT) aircraft configuration, a turbine blade with narrow tip clearance, and a partial-flap wing geometry. The octree-advancing front method is shown to be effective for generating both surface and volume meshes for each of the configurations considered. Results of a viscous flow simulation on the HSCT configuration are also presented as a demonstration of the effectiveness and suitability of the generated hybrid grids.

II. Octree-Advancing Front Method

A combined octree-advancing front method is used to generate the unstructured grid. Advancing front types of methods require specification by the user of the distribution of three parameters over the entire domain to be gridded. These field functions are 1) the node spacing, 2) the grid stretching, and 3) the direction of stretching. By using the octree-advancing front method, these parameters do not need to be specified. Instead, they are determined via an automatically generated octree. There is no need for a special background mesh, which has been the backbone of previous advancing front generators.[3,4] Construction of this background mesh requires user intervention and benefits from the optional solution of a Poisson equation, which can be expensive.[25] The octree, on the other hand, is generated with minimal user input and is an efficient data structure to use to extract the mesh parameters.

The octree data structure is similar to earlier data structures used for search operations during the grid generation process.[26] The divide-and-conquer process starts with a master hexahedron that contains the body. This hexahedron is recursively subdivided into eight smaller hexahedra called octants. Any octant that intersects the body is a boundary octant and is subdivided further (inward refinement). The subdivision of a boundary octant ceases when its size matches the local length scale of the geometry. The choice of the local length scale depends on the particular application of the octree. The length scale can be chosen to be local prism thickness, edge length, or curvature. This flexibility allows the same octree creation technique to be used for many different unstructured applications.

Then, the hexahedral grid is further refined in a balancing process (outward refinement) to prevent neighboring octants whose depth differs by more than one. Outward refinement is performed to ensure that the final octree varies smoothly in size away from the original surface. The sole criterion for outward refinement is a depth difference greater than one between the octant itself and any of its neighbors. The outward refinement continues until no octants meet the refinement criterion. Typically, five sweeps are performed to produce a balanced octree.

Two important features of the octree-advancing front method are its capability to match disparate length scales and its geometry independence. The octree is able to ensure a smooth size transition over the large range of length scales that are present in a viscous mesh. The octree is also able to be used for many different types of geometries with minimal user interaction.

A. Disparate Local Length Scales

Octree refinement is terminated when the size of a boundary octant is the same size as the local length scale of the geometry. This local length scale depends on the current application. Three different applications are considered in the present work; namely, surface mesh generation, tetrahedral mesh generation for hybrid grids, and all-tetrahedral mesh generation.

For surface mesh generation, the local length scale is determined by the local curvature of the geometry. An isotropic surface triangulation is used to approximate the curvature by calculating angles between adjacent faces. The local length scale is proportional to the angle between the faces. This length scale is small in areas where the curvature is large, i.e., the trailing edge of a wing, and large where the geometry is flat. Figure 1 demonstrates the octree for a partial-flap high-lift wing with clustering at midspan for demonstration purposes. The contour lines on the surface correspond to the local length scale of the geometry. Note how the octants intersecting the surface vary in size according to the local length scale and how they also smooth the abrupt size transition from high-curvature regions to flat regions.

The distance between surfaces is another length scale used for surface mesh generation. The local length scale is proportional to this distance. This allows for automatic clustering in regions where surfaces are in close proximity. Figure 2 demonstrates the effect of this clustering. The spheres on the left have a total of 2068 faces and the spheres on the right, which are one-tenth the distance apart, have 3853 faces. Note that the increase in the number of faces takes place only in the narrow gap between the spheres.

For hybrid prismatic/tetrahedral mesh generation, the local length scale is simply the local thickness of the last prismatic layer. This will ensure that the size of the tetrahedra in the direction normal to the outer prismatic surface is the same as the height of the neighboring prisms. This smooth transition in size from the prisms to the tetrahedra is important for accuracy of the numerical method. The

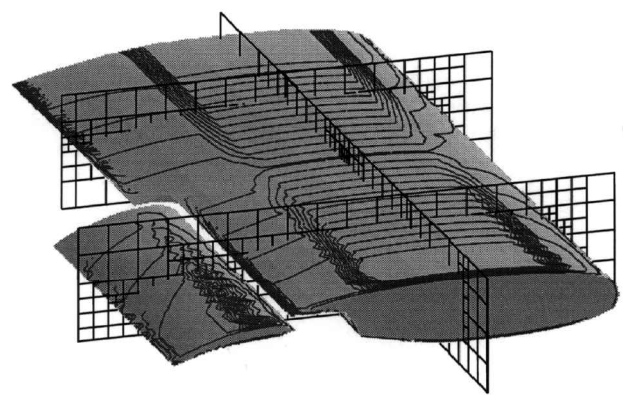


Fig. 1 Octree along three plane cuts for a partial-flap high-lift wing with clustering at midspan. Contour lines correspond to the local length scale of the geometry, which changes abruptly.

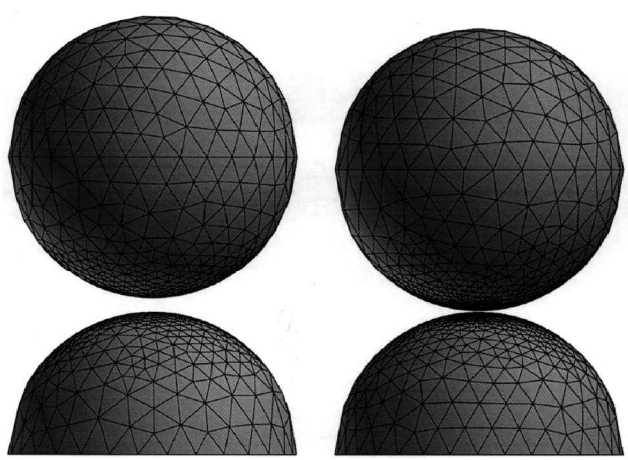


Fig. 2 Spheres in close proximity. Spheres on the left are made up of 2068 faces, and spheres on the right have 3853 faces and are one-tenth the distance apart.

Fig. 3 HSCT configuration with engines.

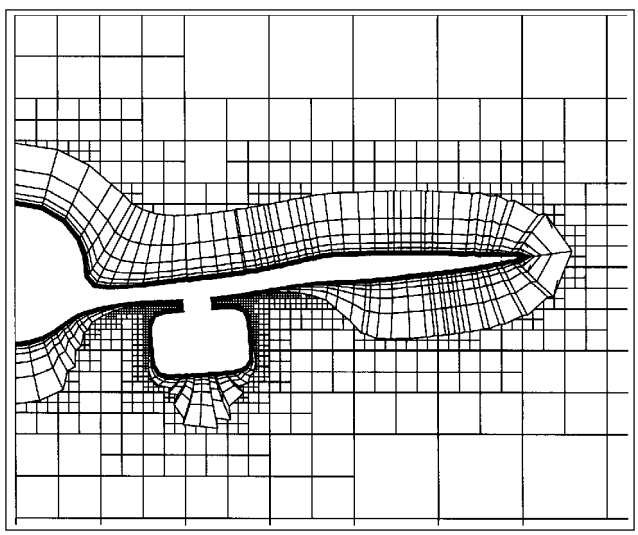


Fig. 4 Plane cut of an octree mesh for the HSCT aircraft that shows how the octree matches the local thickness of the final prismatic layer. (Every third layer of the prismatic mesh is shown.)

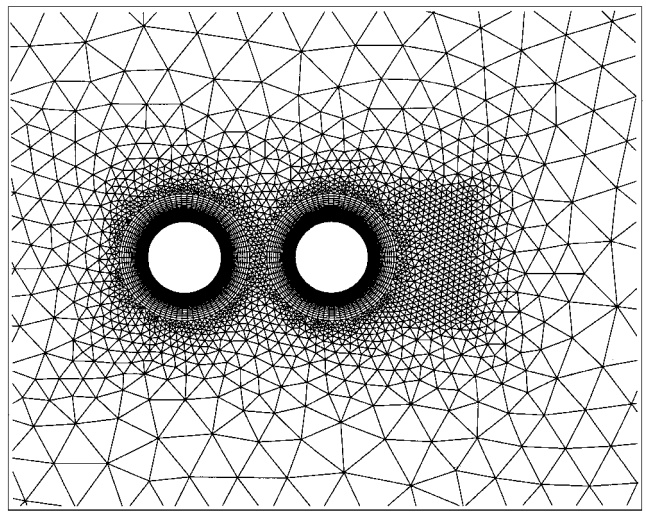


Fig. 5 Two-dimensional mesh for two cylinders in close proximity. Point distribution in the wake region of the second cylinder is caused by a user-specified clustering.

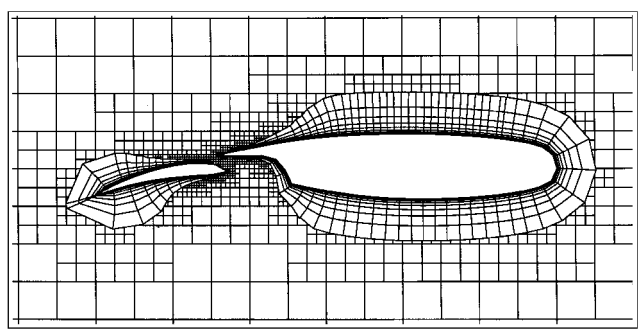


Fig. 6 Plane cut of an octree mesh for the partial-flap high-lift wing at midspan of the flapped region showing how the octant sizes match the local thickness of the final prismatic layer. (Every third layer of the prismatic mesh is shown.)

surface geometry of the HSCT aircraft is shown in Fig. 3. Figure 4 shows a plane cut of an octree mesh for this geometry. Note how the size of the octants intersecting the surface matches the thickness of the last prismatic layer, even in the region near the engine where the last prismatic layer is several orders of magnitude thinner than its size away from the engine. Finally, for an all-tetrahedral mesh, the local length scale is the local edge length of the original triangulated surface.

In any of these mesh generation applications, a user can also specify areas of the mesh for clustering. For instance, if a user knows a priori where a shock will form, points can be clustered in that region. User-specified clustering is optional and all of the length scales previously mentioned will generate an adequate mesh automatically. Figure 5 shows a two-dimensional example of a user-specified clustering in the wake region of two tandem cylinders.

B. Geometry Independence of the Octree

The geometry independence of the octree is demonstrated through its application to both the HSCT aircraft with an engine and the partial-flap wing. Each of these geometries is complex and demonstrates the capability of the octree in adapting to different geometry features. The HSCT aircraft has a cavity between the engine and the wing; the partial-flap wing has two different thin gaps between the flap and the wing. Figure 4 shows a plane cut of the octree around the engine of the HSCT. The octants in the cavity between the wing and the engine are much smaller, and they match the local thickness of the outer prismatic layer. Note that, for clarity, only every third layer of the prismatic mesh is shown.

Figure 6 shows a plane cut of the octree around the partial-flap high-lift wing taken at midspan of the flapped region. The octants in the gap between the flap and the main airfoil are smaller than those out of the gap and match the local thickness of the outer prismatic layer accurately. Note that, for clarity, only every third layer of the prismatic mesh is shown.

Note that the octree for each of these cases was generated with minimal user interaction.

C. Octree Guides Advancing Front Mesh Generation

The advancing front volume grid generation starts from the surface of the body or the outermost prismatic surface for the case of a hybrid grid. The triangular faces of this surface form the initial front list. A face from this list is chosen to start the tetrahedra generation. Then, a list of points is created that consists of a new node as well as of nearby existing points of the front. One of these points is chosen to connect to the vertices of the face. Following the choice of the point, a new tetrahedron is formed. The list of the faces, edges, and points of the front is updated by adding and removing elements. The algorithm followed in the present work is the one presented in Ref. 4. The method requires a data structure that allows for efficient addition/removal of faces, edges, and points as well as for fast identification of faces and edges that intersect a certain region. The alternating digital tree algorithm is employed for these tasks.[27]

The tetrahedra that are generated by this octree method grow in size as the front advances away from the original surface. Their size, the rate of increase of their size, and the direction of the increase are all given from the octree. The octants are progressively larger with distance away from the body. Their sizes determine the characteristic size of the tetrahedra that are generated in their vicinity. This method is flexible and can be used to generate tetrahedra around different types of geometry.

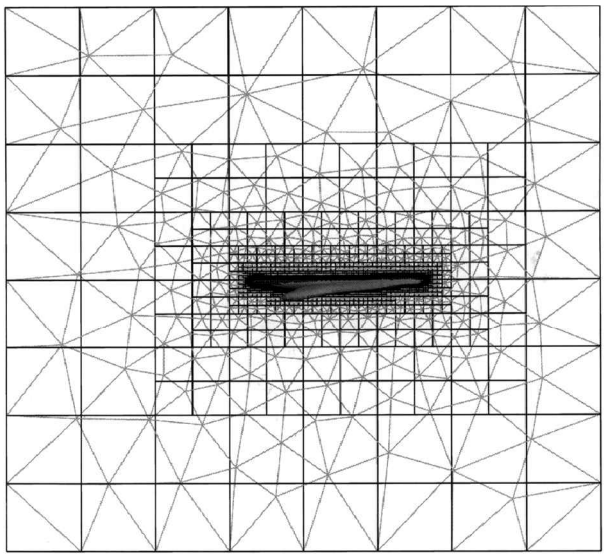


Fig. 7 Effect of the octree on growth of the tetrahedra for the HSCT aircraft geometry. View of the octants (quadrilateral faces) as well as of the tetrahedra (triangular faces) on the symmetry plane. Growth of the tetrahedra away from the outermost prism surface follows growth of the octree quite faithfully.

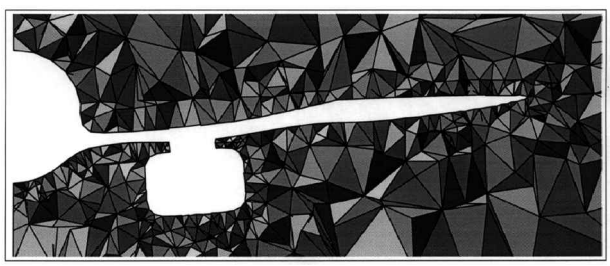


Fig. 8 Field cut of a tetrahedral mesh for the HSCT aircraft generated by the octree-advancing front method.

Figure 7 illustrates the symmetry plane of the HSCT configuration without the engine. The quadrilaterals (dark lines) correspond to the faces of the octants on this plane, and the triangles (light lines) correspond to the faces of the tetrahedra. It is observed that the size of the tetrahedra, as well as the stretching of the mesh and the direction of stretching, is guided quite accurately by the octree.

The octree-advancing front method can also be used to create meshes for inviscid simulations. Given an initial surface triangulation, the octree is refined until the boundary octants match the size of the local surface triangulation. Figure 8 shows a field cut of an all-tetrahedral mesh around the HSCT with engines. The octree has effectively matched the edge lengths of the surface triangulation and smoothly guided the sizes of the tetrahedra generated.

The surface mesh generation proceeds in the same manner as the tetrahedral mesh generation, except that surface triangles are generated from an initial front made up of edges.[4] The surface geometry is treated as a patchwork of panels. Structured surface grids are used to approximate the surface. Ferguson patches are then formed from these structured grids. A conversion tool has been written to automatically convert Initial Graphics Exchange Specification (IGES) format CAD files to the structured grids for use in the surface grid generator. Boundary segments along the sides of each panel are discretized first by using the octree to control the point distribution. The interior of each panel is then filled with triangles, using the same octree for each panel to ensure smooth size transitions across panel boundaries. New triangles are generated by using either already existing points or new points generated on the surface with information from the octree. The octree allows for a smooth transition in size on the surface from areas where the triangles are small (i.e., trailing edge) to areas where the triangles are larger. Figure 9 shows how

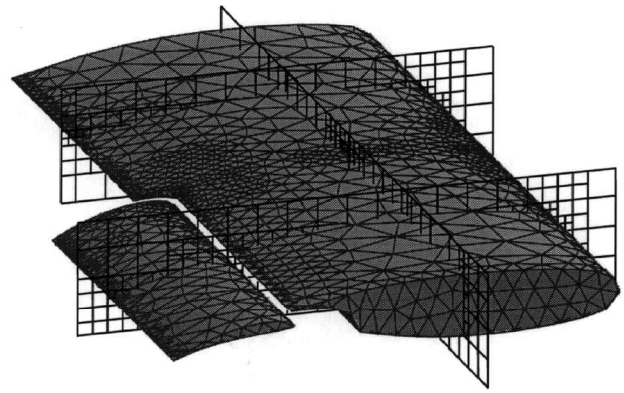


Fig. 9 Octree along three plane cuts for a partial-flap wing with user-specified clustering at midspan and the corresponding triangular mesh generated by using this octree.

the octree guides the size of the triangular mesh for the partial-flap high-lift wing. Note how the triangular mesh is much more dense near the leading and trailing edge of the flap where the octants are smaller. Also, note that this is the same octree shown in Fig. 1. These two figures show how the local length scale information is used to generate the octree and how the octree is used to generate a surface triangulation. The generated surface mesh matches the local length scales but has a smooth transition in size from the dense areas of the mesh to the coarse areas.

The advancing front method creates a new element by connecting each face or edge of the current front to either a new or an existing node. This new point is found by using a characteristic distance, δ, which is calculated from the size of the local octant to which the face of the front belongs. Specifically,

$$\delta = \alpha s t^{l_t - l} \tag{1}$$

where α is a scaling factor, st is the stretching parameter, l_t is the total number of octant levels, and l is the level of the local octant. The value of st controls the rate of growth of the mesh. The lower the value of st, the less the mesh increases in size away from the body. A typical value of the stretching parameter st is 1.8. The level l of the local octant is the number of subdivisions of the master octant required to get to the size of the local octant.

For hybrid mesh generation, smooth transition in size from the prisms to the tetrahedra is important for accuracy of the numerical methods. The value of the scaling factor α is calculated so that the initial marching size (δ) of the tetrahedra equals the local thickness of the outermost prismatic layer. This feature allows smooth transition in size from the last layer of prisms into the tetrahedral region.

For surface mesh generation, α can be varied to generate different meshes with the same octree. Higher values of α result in coarser meshes with the same size variation, and lower values of α yield finer meshes. Figure 10 shows two different meshes for the HSCT generated by using the same octree to control the spacing but different values of α. The top mesh has 4401 triangular faces, and the bottom mesh has 20,106 faces. Note that even with 4401 faces, all of the geometry details have been captured.

D. Anisotropic Surface Meshes

The octree-advancing front method can also create anisotropic surface meshes. Anisotropic meshes are useful in reducing the number of triangular faces needed to capture all the flow features in a simulation. Allowing high-aspect ratio triangles aligned with geometry and flow features in regions that exhibit strong directionality enables a substantial savings in number of both surface and volume grid elements. A user needs only to specify the following: 1) a line segment that defines the direction of stretching of the mesh, 2) the aspect ratio AR of the triangles desired along that line segment, and 3) the area of influence $d_{\max}$ of the line segment. Examples of such line segments include the leading edges, trailing edges, and engine inlets. The method for generating anisotropic meshes starts with the size δ_{oct} given by the octree and augments it with the perpendicular

Fig. 10 Two surface meshes for the HSCT aircraft generated by using the same octree. Top mesh has 4401 triangular faces, and bottom mesh has 20,106 faces.

distance d from the user-specified line segment. The local mesh size is now characterized by three sizes, δ_1, δ_2, and δ_3, given by

$$\delta_1 = c \times \delta_{\text{oct}} \qquad \delta_2 = \delta_{\text{oct}} \qquad \delta_3 = \delta_{\text{oct}} \tag{2}$$

with

$$c = [(AR - 1)/d_{\max}]d + AR \tag{3}$$

where δ_1 is the size of the mesh in the direction of the line segment and δ_2 and δ_3 are the sizes of the mesh in directions perpendicular to the line segment and perpendicular to each other. The method is flexible and robust, using multiple line segments at different locations and directions to define directionality on different parts of the surface. Furthermore, it provides a smooth transition between regions of different directionality.

Figure 11 shows a surface mesh on the ONERA M6 wing with two user-defined line segments: one on the inboard half of the leading edge with $AR = 10$ and another on the outboard half of the trailing edge with $AR = 5$. Note the smooth transition from the anisotropic regions to the isotropic areas between the two areas of high AR faces. Furthermore, the octree yielded smaller faces at the leading edge compared to the trailing edge. Use of high AR faces results in a significant reduction in the number of triangles and tetrahedra. Figure 12 shows both an isotropic surface mesh for the M6 wing and an anisotropic mesh created with line segments extending over the entire leading and trailing edges with the same aspect ratios as the previous mesh. The isotropic mesh has 39,290 faces, and the anisotropic mesh has 6333 faces while maintaining the same chordwise point density obtained from the same octree. These meshes show the 6.2:1 reduction in number of generated faces when an anisotropic method is used. This reduction in faces leads to a substantial reduction in the number of elements of the corresponding volume mesh.

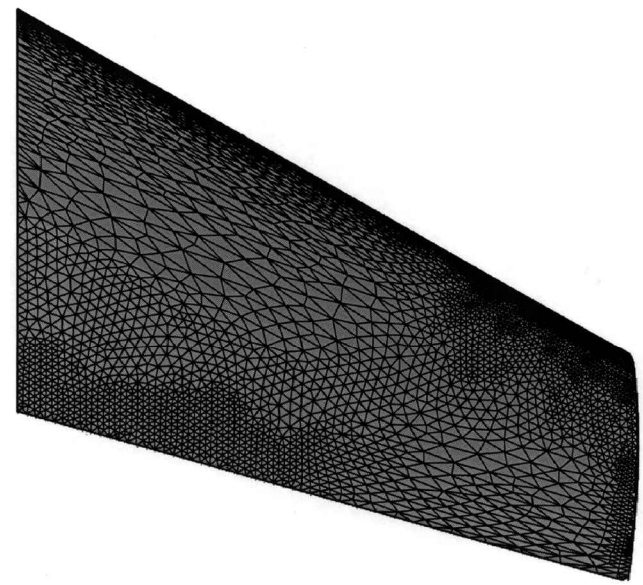


Fig. 11 Two staggered regions of directional mesh on the surface of the ONERA M6. The first line segment is along the inboard half of the leading edge, and the second is along the outboard half of the trailing edge. Note the smooth transition between anisotropic and isotropic regions in the mesh.

E. Automatic Partial Remeshing

Grids generated by an advancing front type scheme can contain regions of low quality within the mesh domain. These low-quality regions must be altered before the mesh can be used with a flow solver. A method for improving low-quality regions has been developed. This method removes low-quality regions from the mesh and fills the resulting cavities by using the same advancing front generator on the new front defined by the surface of these holes.

To properly define the low-quality regions of the mesh, the quality of a given region must be quantified. The measure used in the present work is the volume ratio of the two tetrahedra sharing each face, $R = \text{Vol}_{\max}/\text{Vol}_{\min}$. Large values of R indicate a very stretched mesh. If $R = 1$, the mesh is locally uniform.

Once the low-quality regions of the mesh have been located by the quality measure R, these regions must be removed from the mesh. For each face with a value of R greater than a user-specified value R_{sp}, a cavity is opened around the low-quality region by removing tetrahedra. The radius of the opened cavity is dependent on the local length scale of the mesh.

After cavities have been formed around each of the low-quality regions of the mesh, the exposed triangular faces inside the cavities are put together to form a new initial front. Then, the advancing front generator refills the cavities with better quality tetrahedra. This process of cavity definition and cavity remeshing is repeated until a specific level of quality is reached.

The entire process of cavity definition and remeshing is performed automatically with no user intervention. The remeshing process is efficient and typically takes a quarter of the time that the initial tetrahedral generation requires.

Figure 13 shows the number of faces left in the mesh with a value of $R > 50$ for a HSCT aircraft without engines. Initially, 230 faces

Fig. 12 Significant savings in number of triangles are realized because of the use of leading and trailing edge line segments for the M6 wing. The top mesh is an isotropic mesh with 39,290 faces. The bottom mesh is an anisotropic mesh with 6333 faces. Note that even though the isotropic mesh has six times the number of faces, the anisotropic mesh has the same chordwise point distribution.

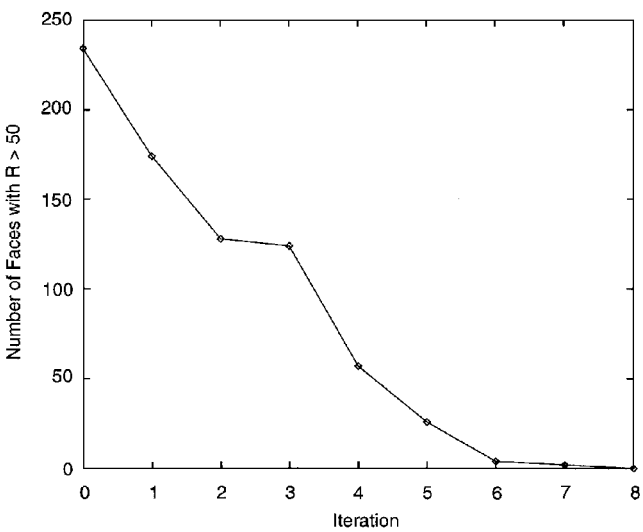


Fig. 13 Elimination of areas with large variation in local cell sizes for the HSCT aircraft without engines. Number of faces with volume ratio $R > 50$ left after each local remeshing.

had $R > 50$. After eight remeshing iterations, no faces with $R > 50$ were left. The graph shows that the automatic partial remeshing method was effective in eliminating these low-quality regions from the mesh.

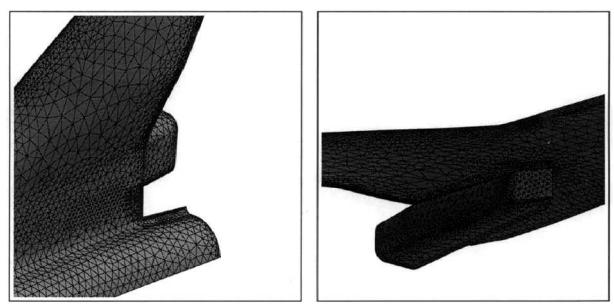
III. Applications of the Octree-Advancing Front Method

Surface meshes and tetrahedral grids were generated for a variety of geometries to demonstrate the effectiveness of the developed octree-advancing front grid generator. The geometries presented here include the HSCT aircraft configurations with and without engines and a turbine blade with narrow tip clearance. A viscous flow simulation is presented over the HSCT configuration without engines to demonstrate the validity of the generated grids. The finite volume scheme used for the numerical simulations uses central space differencing and Lax–Wendroff time marching. Scheme operations are cast in edge-based form. The details regarding the Navier–Stokes solver are presented in Ref. 28.

A. Surface Meshes

The octree-advancing front method was used to generate different surface meshes for the HSCT configuration with engines. Figure 10 shows two different meshes with different levels of coarseness. The top mesh has 4401 triangular faces, and the bottom mesh has 20,106 faces. The same octree was used to generate both meshes. The only parameter varied was α, the scaling parameter in Eq. (1). Figure 14 shows closeups of various regions of the 20,106-face mesh. Each of these views shows how the octree-advancing front method was able to generate small faces in areas of high curvature and was thus able to resolve the details of the geometry with a reasonable number of faces. In both cases, the smooth transition the octree provides from small faces in areas of high curvature to large faces in flat regions of the surface.

Figure 15 shows an anisotropic surface mesh for the HSCT with a flow-through engine. The anisotropic mesh has 30,189 faces, whereas a similar isotropic case has 60,583 faces. The anisotropic



Wing–body junction **Engine region**

Fig. 14 Closeup views of a surface mesh for the HSCT configuration with engines. The mesh shown has 20,106 faces.

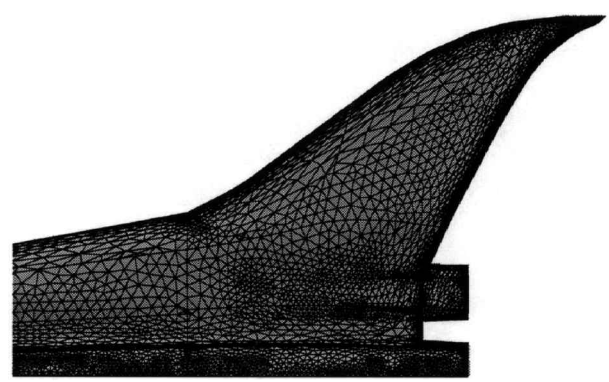


Fig. 15 Anisotropic surface mesh for the HSCT with flow-through engines. Anisotropic regions near the leading and trailing edge of the wing are shown. The mesh has 30,189 faces; a similarly spaced isotropic mesh would have 60,583 faces.

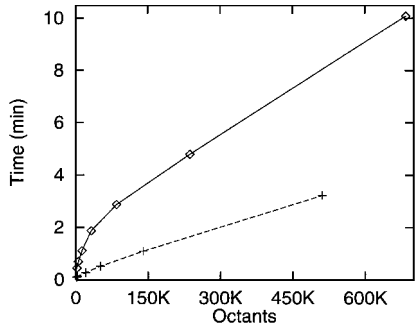


Fig. 16 Linear scaling of the octree generation time with number of octants: ——, HSCT aircraft surface and – – –, partial-flap wing surface.

mesh has reduced the number of faces by a 2:1 ratio. The figure shows how the use of anisotropic line segments reduces the number of points on the wing in the spanwise direction while maintaining the same number of points in the chordwise direction. The octree generates the fine mesh required in the chordwise direction, whereas anisotropic line segments allow for a larger lateral mesh spacing; hence, the octree reduces the number of faces required in this region of the mesh.

The generation of the octree for each of the previous cases is a fast process that scales linearly with the number of octants generated. Figure 16 shows how the time varies with number of octants generated for the both the HSCT aircraft and the partial-flap wing surface. The figure shows a linear relationship between time to generate the mesh and number of octants generated. It also shows that the overall time it takes to generate a fine octree is on the order of minutes. The timings were performed on an IBM RS/6000 model 390 workstation.

B. Hybrid Meshes

The suitability of the octree-advancing front method for hybrid meshes is demonstrated through its application to different geometries. The method is used to generate the tetrahedral part of hybrid prismatic/tetrahedral meshes for both the HSCT aircraft with and without engines and a turbine blade with narrow tip clearance. The tetrahedral part of each of these hybrid grids was generated by using the octree-advancing front method, whereas the prismatic meshes were generated by using the method described in Refs. 16 and 17. Automatic partial remeshing was applied to all hybrid meshes shown in this section. A Navier–Stokes flow simulation is shown around the HSCT aircraft to demonstrate the validity of the hybrid mesh for viscous flow calculations.

The first case that is considered to test the robustness and universality of the developed octree-advancing front grid generator is the HSCT configuration with engines. Figure 3 shows a view of the initial surface. The surface was composed of 20,784 triangular faces. The hybrid grid consisted of 39 layers of prisms (810,576 prismatic cells) and 487,854 tetrahedral cells.

An isometric view of the hybrid grid is shown in Fig. 17. The figure shows the mesh on two planes that are perpendicular to each other. The first plane is that of the symmetry, which shows the quadrilateral signature of the prisms along with the triangular signature of the tetrahedra. The second plane is a field cut along the fuselage that illustrates the three-dimensional hybrid nature of the grid. The structure of the prisms and the smooth transition from prisms to tetrahedra are worth observing. The size of the elements varies smoothly at the prism/tetrahedra interface in the region of the engine cavity. Figure 18 is a field cut along the fuselage that better illustrates the cavity region and the smooth variation in prismatic cell thickness away from the cavities. It shows that the tetrahedra match the local prism cell sizes relatively well even in the cavity where local length scales are as much as 200 times as small as those away from the cavity.

The tetrahedra generation process takes on the order of 30 min for a typical geometry. Figure 19 shows the time required to generate tetrahedra for a hybrid mesh for the HSCT without engines. To generate the different meshes, the same octree was used and the value of the stretching parameter, st in Eq. (1), was varied. The

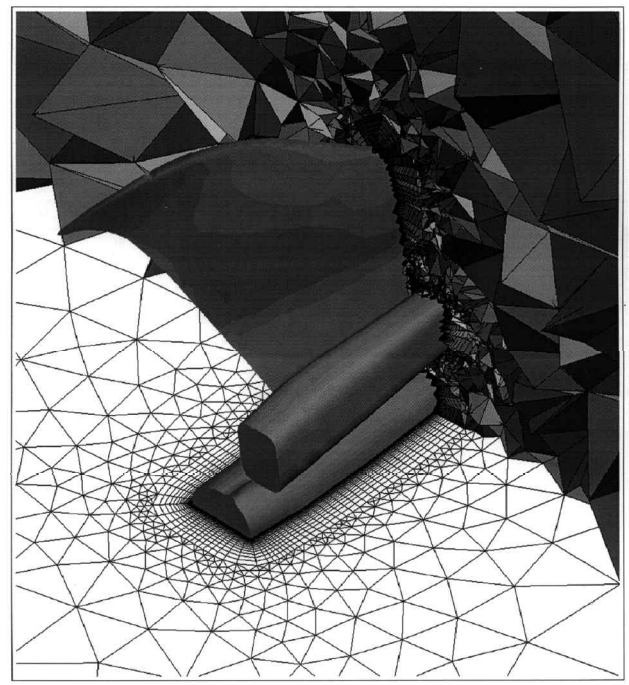


Fig. 17 View of the hybrid mesh around the HSCT aircraft with engines on two different planes that are perpendicular to each other. The first plane is that of the symmetry, and the second is a field cut intersecting the fuselage and engine.

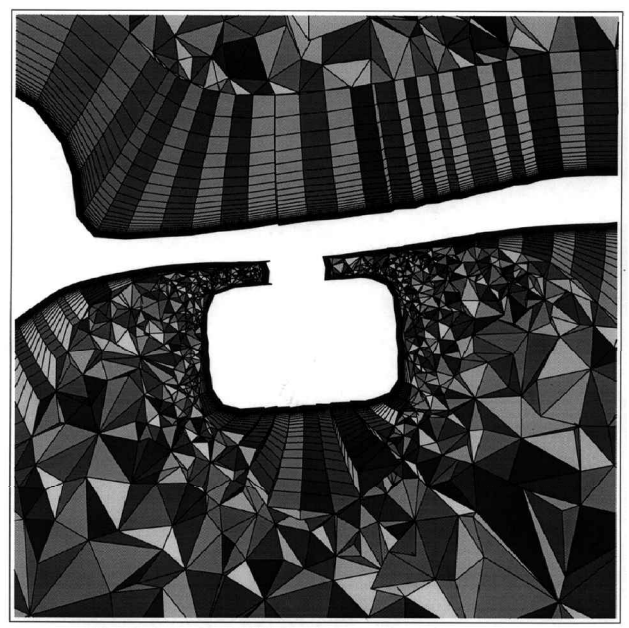


Fig. 18 Closeup of the hybrid grid for the HSCT aircraft around the engine cavity. The tetrahedral mesh is very dense here compared to other regions so as to match the thin local prism cell sizes.

figure shows that the time to generate tetrahedra scales linearly even when 500,000 tetrahedra are generated. These timings were performed on the same IBM model 390 workstation.

Supersonic flow around the HSCT without engines at $M = 3.0$ and a $Re = 10^6$ is considered to check the validity of the method for generating meshes for flow simulations. The hybrid grid used in the simulation consists of 176,480 prisms and 171,483 tetrahedra. Generation of the prisms took about 90 s on an IBM 390 workstation; generation of the tetrahedra took about 27 min on the same machine. A view of the entropy contours along with the hybrid mesh for the HSCT is shown in Fig. 20. The cut has been taken along the axis perpendicular to the fuselage ($x = 180$). Note that the irregularity of

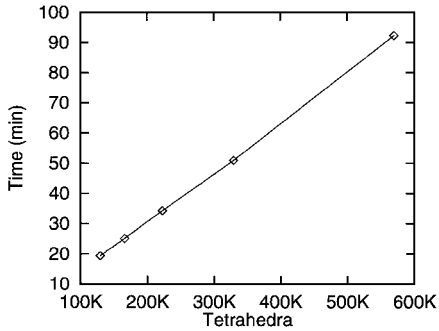


Fig. 19 Linear scaling of the tetrahedra generation time with number of tetrahedra for the HSCT aircraft.

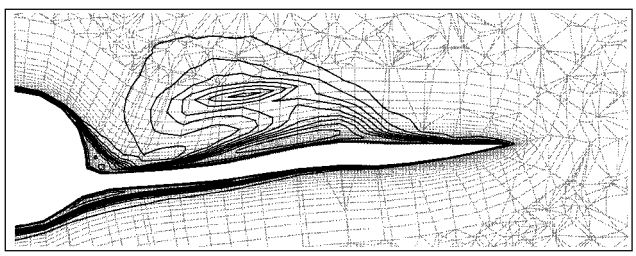


Fig. 20 Plane cut along the fuselage showing the hybrid mesh as well as the corresponding entropy contours. The hybrid grid interface has no adverse effects on the accuracy of the solution. Case of supersonic flow ($M_\infty = 3.0$, $Re = 10^6$, $\alpha = 5.0$ deg) around the HSCT aircraft.

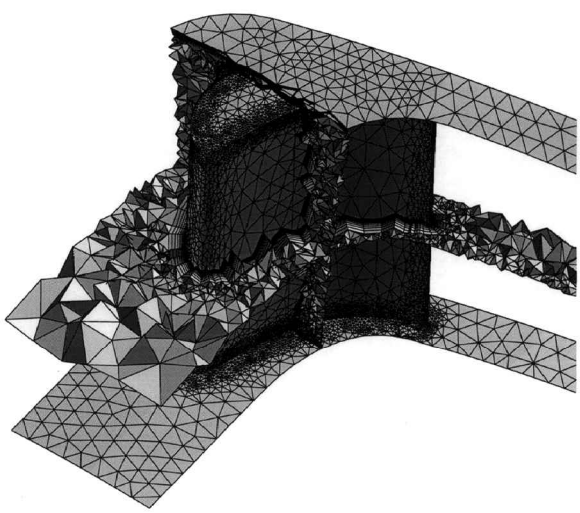


Fig. 21 Field cuts of the hybrid mesh around a turbine blade with narrow tip clearance. The two field cuts are on perpendicular surfaces intersecting the blade.

the mesh is due to the fact that the grid is not coplanar with the cut. The recirculation region is clearly visible at the junction between the wing and the fuselage. Furthermore, a qualitative assessment of the contour lines indicates that the numerical solution is not affected by the transition from prisms to tetrahedra because there are no abrupt changes in the contours at the interface.

The final case considered to demonstrate the robustness and universality of the developed octree-advancing front grid generator is a turbine blade with narrow tip clearance. The surface was composed of 13,663 triangular faces. The hybrid grid consisted of 14 layers of prisms (191,282 prismatic cells) and 415,086 tetrahedral cells. Figure 21 shows two perpendicular field cuts of the hybrid mesh around the blade in this geometry. The tetrahedra were able to easily match the prismatic thickness everywhere, including the small gap between the tip of the blade and the shroud. Note that the octree-advancing front method was able to mesh an internal geometry as easily as the external geometries presented earlier.

IV. Concluding Remarks

The octree-advancing front method provided an automatic method for generating unstructured meshes. The method was effective in generating surface triangulations for different complex geometries including the partial-flap wing and the HSCT aircraft. The octree allowed surface triangulations to be generated that captured all of the geometry features with a minimum number of faces. The octree also provided for a smooth variation of grid size over the entire surface mesh. The time required for octree generation was proportional to the number of elements generated.

Anisotropic surface meshes were generated by using the octree and minimal user input. These anisotropic meshes resulted in a 6.2:1 reduction in the number of faces generated. Smooth transition between the different regions of directionality was also accomplished.

Generation of tetrahedra via the advancing front method was also made simpler and more automatic by eliminating the traditional user-defined background mesh for determination of mesh spacing. An automatically generated octree guided the growth of the tetrahedra and enabled a smooth transition of the mesh from the prisms to the tetrahedra in a hybrid mesh. The universality of the octree-advancing front method was demonstrated through its application to different complex geometries: the partial-flap high-lift wing, a turbine blade with a narrow tip clearance, and the HSCT aircraft configuration with and without engines. The HSCT aircraft configuration demonstrated that the method is flexible enough to adapt to 200:1 size variations in the local length scale. Local remeshing of the tetrahedral mesh proved very effective in removing areas of abrupt changes in sizes of the tetrahedra. The time required for generation of tetrahedra was proportional to the number of elements generated.

Acknowledgments

This work was supported by NASA Grant NAG1-1459, National Science Foundation Grant ASC-9357677 (NYI Program), and Texas Advanced Technology Program Grant 003658-413. Computing time was provided by the NAS Division of NASA Ames Research Center as well as by the High Performance Computing Facility of the University of Texas. The authors wish to thank Javier Garriz of NASA Langley Research Center's Geolab for his assistance with the HSCT surface geometry.

References

[1]Thompson, J. F., and Weatherill, N. P., "Aspects of Numerical Grid Generation: Current Science and Art," AIAA Paper 93-3539, June 1993.

[2]Baker, T. J., "Developments and Trends in Three-Dimensional Mesh Generation," Transonic Symposium, NASA Langley Research Center, Hampton, VA, April 1988.

[3]Löhner, R., and Parikh, P., "Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method," AIAA Paper 88-0515, Jan. 1988.

[4]Peraire, J., Morgan, K., and Peiro, J., "Unstructured Finite Element Mesh Generation and Adaptive Procedures for CFD," *Application of Mesh Generation to Complex 3-D Configurations*, AGARD Conf. Proceedings 464, June 1990, pp. 18.1–18.12.

[5]Yerry, M. A., and Shephard, M., "Automatic 3-D Mesh Generation by the Modified Octree Technique," *International Journal of Numerical Methods in Engineering*, Vol. 20, 1984, pp. 1965–1990.

[6]Baker, T. J., "Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation," *Engineering with Computers*, Vol. 5, 1989, pp. 161–175.

[7]Holmes, D. G., and Snyder, D. D., "The Generation of Unstructured Meshes Using Delaunay Triangulation," Numerical Grid Generation in Computational Fluid Mechanics, *Proceedings of the Second International Conference in Numerical Grid Generation in Computational Fluid Dynamics* (Miami, FL), edited by S. Sengupta, J. Hauser, P. R. Eisman, and J. F. Thompson, Pineridge, Swansea, Wales, UK, 1988, pp. 315–333.

[8]Weatherill, N. P., "Delaunay Triangulation in Computational Fluid Dynamics," *Computers and Mathematics with Applications*, Vol. 24, No. 5/6, 1992, pp. 129–150.

[9]Marcum, D. L., and Weatherill, N. P., "Unstructured Grid Generation using Iterative Point Insertion and Local Reconnection," *Proceedings of the AIAA 12th Applied Aerodynamics Conference* (Colorado Springs, CO), AIAA, Washington, DC, 1994, pp. 403–420 (AIAA Paper 94-1926).

[10]Lo, S. H., "A New Mesh Generation Scheme for Arbitrary Planar Domains," *International Journal of Numerical Methods in Engineering*, Vol. 21, No. 8, 1985, pp. 1403–1426.

[11] Shephard, M. S., Guerinoni, F., Flaherty, J. E., Ludwig, R. A., and Baehmann, P. L., "Finite Octree Mesh Generation for Automated Adaptive Three-Dimensional Flow Analysis," *Proceedings of the Second International Conference on Numerical Grid Generation in Computational Fluid Mechanics* (Miami, FL), edited by S. Sengupta, J. Hauser, P. R. Eisman, and J. F. Thompson, Pineridge, Swansea, Wales, UK, 1988, pp. 709–718.

[12] Aftosmis, M., Berger, M., and Melton, J., "Adaptation and Surface Modeling for Cartesian Mesh Methods," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference* (San Diego, CA), AIAA, Washington, DC, 1995 (AIAA Paper 95-1725).

[13] Mavriplis, D. J., "An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness," AIAA Paper 93-0671, Jan. 1993.

[14] Merriam, M. L., "An Efficient Advancing Front Algorithm for Delaunay Triangulation," AIAA Paper 91-0792, Jan. 1991.

[15] Muller, J. D., Roe, P. L., and Deconinck, H., "A Frontal Approach for Node Generation in Delaunay Triangulations," *Unstructured Grid Methods for Addiction Dominated Flows*, VKI Lecture Notes, 9-1, 9-7, AGARD Publication R-787, 1992.

[16] Kallinderis, Y., Khawaja, A., and McMorris, H., "Hybrid Prismatic/Tetrahedral Grid Generation for Complex Geometries," AIAA Paper 95-0211, Jan. 1995.

[17] Khawaja, A., McMorris, H., and Kallinderis, Y., "Hybrid Grids for Viscous Flows Around Complex 3-D Geometries Including Multiple Bodies," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference* (San Diego, CA), AIAA, Washington, DC, 1995 (AIAA Paper 95-1685).

[18] McMorris, H., and Kallinderis, Y., "Octree-Advancing Front and Automatic Partial Remeshing for Tetrahedra Generation," *Proceedings of the AIAA 12th Computational Fluid Dynamics Conference* (San Diego, CA), AIAA, Washington, DC, 1995 (AIAA Paper 95-1762).

[19] Kallinderis, Y., Khawaja, A., and McMorris, H., "Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flows Around Complex Geometries," *AIAA Journal*, Vol. 34, No. 2, 1996, pp. 291–298.

[20] Kallinderis, Y., "Prismatic/Tetrahedral Grid Generation for Complex Geometries," *VKI Lecture Notes*, Twenty-Seventh Computational Fluid Dynamics Lecture Series, Brussels, Belgium, March 1996.

[21] Pirzadeh, S., "Structured Background Grids for Generation of Unstructured Grids by Advancing Front Method," *AIAA Journal*, Vol. 31, No. 2, 1993, pp. 247–265.

[22] Pirzadeh, S., "Progress Toward a User-Oriented Unstructured Viscous Grid Generator," AIAA Paper 96-0031, Jan. 1996.

[23] Peraire, J., and Morgan, K., "Viscous Unstructured Mesh Generation Using Directional Refinement," *Proceedings of the International Conference on Numerical Grid Generation in Computational Field Simulations* (Mississippi State, MS), National Science Foundation Research Center for Computational Field Simulation, 1996.

[24] Hufford, G. S., and Mitchell, C. R., "The Generation of Hybrid and Unstructured Grids Using Curve and Area Sources," AIAA Paper 95-0215, Jan. 1995.

[25] Pirzadeh, S., "Recent Progress in Unstructured Grid Generation," AIAA Paper 92-0445, Jan. 1992.

[26] Löhner, R., "Some Useful Data Structures for the Generation of Unstructured Grids," *Communications in Applied Numerical Methods*, Vol. 4, No. 1, 1988, pp. 123–135.

[27] Bonet, J., and Peraire, J., "An Alternating Digital Tree (ADT) Algorithm for Geometric Searching and Intersection Problems," *International Journal of Numerical Methods in Engineering*, Vol. 31, No. 1, 1990, pp. 1–17.

[28] Parthasarathy, V., Kallinderis, Y., and Nakajima, K., "A Hybrid Adaptation Method and Directional Viscous Multigrid with Prismatic/Tetrahedral Meshes," AIAA Paper 95-0670, Jan. 1995.

C. R. Mitchell
Associate Editor